

Creació d'una aplicació Android d'una xarxa social

Alumne: Joan Fàbregas Tell
Director: Juan José Fábregas
11 setembre de 2016

Introducció

Aquest Treball Final de Grau va néixer a proposta del propi director d'aquest treball i en el que vaig trobar l'excusa perfecte per tal de poder aprendre els coneixements d'Android que estava buscant per incorporar al meu currículum. Es tracta de la creació d'un prototip amb Android (concretament l'API 23 referent a l'Android 5.1) d'una xarxa social per comentar i valorar sèries. Paral·lelament a la realització d'aquesta aplicació el treball se suma a un projecte en conjunt amb dos companys més, amb l'Eduard Vallvé i la Laura Nelly, en el que el primer realitza la plana web i la Laura fa els dissenys.

Per a la realització d'aquest treball va ser necessari aplicar un mètode de treball àgil per tal de poder anar al dia i que es treballés de manera sincronitzada. I així aconseguir uns coneixements, els de metodologies de treball àgil que no havíem après de forma extensa durant la carrera però que es fa servir i es dona molta importància en el món laboral.

Altrament, aquest treball m'ha obligat a aprendre un llenguatge de programació i una plataforma de la qual no estava gens acostumat i ha obligat a dedicar moltes hores a la realització de tutorials i cursos per tal de poder acabar aquesta app.

Així doncs aquest treball és el cúmul de moltes reunions pre-desenvolupament amb els companys per tal de definir com havia de ser l'aplicació, planificar i estudiar com serien els hipotètics usuaris que utilitzarien la nostra aplicació i també fer un petit estudi de mercat per saber què es fa i què es deix de fer. Amb tot, comença una idea que es va començar a fer realitat tant bon punt i seguint moltes hores de tutorials i recursos d'internet vaig estar preparat per començar a programar.

Finalment i amb la presentació d'aquest document han acabat 6 mesos de planificacions i desenvolupaments que han donat com a resultat un prototip que malauradament no pot incorporar el disseny desitjat però que funcionalment seria quasi com el resultat final.

Sumari

Introducció	2
Objectius del treball	4
Objectius específics	4
Motivacions	4
Estat de l'art / Context	6
Metodologia	8
Planificació	9
Eines	11
Pressupost	12
Màrqueting	13
Aplicació (TFG)	13
Marca personal	13
Desenvolupament	14
Android	14
Base de dades	15
Programació de l'aplicació	17
Programació estil visual i texts	33
Testeig final	35
Conclusions	35
Bibliografia	36
Artícles	36
Curs complet d'android	36
Enllaços de suport	36

Objectius del treball

L'objectiu final d'aquest treball és el desenvolupament d'un prototip avançat a nivell gràfic i funcional (amb codi) d'una aplicació per a smartphones Android amb la qual es permetrà comentar i valorar sèries. Aquesta aplicació serà la homòloga de la seva versió web, que la farà el company Eduard Vallvé, però amb un disseny específic, que vindrà donat per la companya Laura Nelly.

Objectius específics

- Per tal d'aconseguir-ho hauré **d'aprendre el llenguatge de programació Java/Android** per a poder desenvolupar aquesta aplicació nativa pel sistema Android.
- Per aconseguir una experiència d'usuari òptima l'aplicació s'haurà de **sincronitzar amb una base de dades** (per tenir els últims comentaris i actualitzacions)
- Aprendre a **treballar en equip amb la utilització de una metodologia de treball àgil** anomenada SCRUM.
- Agafar experiència en el prototipat d'aplicacions mitjançant el programa Axure.???

Motivacions

La motivació principal per la qual faig aquest projecte es la de aprendre a desenvolupar una aplicació per a smartphones Android, ja que en cursos anteriors al CITM existia una optativa d'Android a quart, però justament aquest any la van retirar i no l'he pogut fer, i veig en el treball de final de grau una oportunitat excel·lent per a aprendre el llenguatge Java i crear aplicacions per al sistema operatiu Android.

També trobo molt útil el tema del projecte conjunt amb dos companys més, ja que és una molt bona oportunitat per posar en pràctica la metodologia àgil que s'utilitza actualment al món laboral, en aquest cas l'SCRUM amb el KANBAN adaptat al nostre projecte.

Problema

Quan vam començar a plantejar-nos el projecte ens vam adonar que totes les pàgines web que es podrien semblar al que volem fer es quedaven en el simple fet de que l'usuari pogués seguir les seves sèries/pel·lícules preferides i posar un comentari qualsevol així com una puntuació a la sèrie.

Totes les webs analitzades seguien el mateix patró i vam detectar la mancança d'aconseguir que sigui una xarxa social real i que no es limiti a que els usuaris puguin seguir a altre gent sense cap mena d'utilitat real o la de deixar anar un comentari que es perdrà allà en el temps sense cap mena de gràcia.

El que proposem es que els usuaris segueixin als seus propis amics, els que queden cada tarda al bar, els que queden per jugar a jocs online o simplement els amics d'oficina, aquells amb els que es molt probable que s'iniciï una discussió en els comentaris dels usuaris i que a part també es pugui interactuar de forma directa (o a través de xarxes socials com twitter) amb els actors i/o les cadenes de televisió promotores de la sèrie, ja que les sèries que tenen molta activitat a les xarxes socials també són les que veu més gent (Reina Pérez, 2014), i així aconseguir una interacció molt més útil a la par que divertida de l'usuari.

TFG: Creació d'una aplicació Android d'una xarxa social

A part, també incorporar també dinàmiques de joc (PLEX) per tal que l'usuari es diverteixi amb l'aplicació i fer que la torni a utilitzar.

Estat de l'art / Context

Actualment, la forma en què es miren les sèries és la de mirar-les mitjançant els serveis de vídeo sota demanda (VOD), ja sigui des de la mateixa empresa de televisió (exemple Atresmedia amb el seu atresplayer.com) o mitjançant plataformes digitals com Netflix, Yomvi, Wuaki, etc..

Això comporta per una banda que les persones amb una connexió a internet tinguin accés a totes les sèries, i si accedeixen a pàgines específiques poden accedir també a sèries que s'emeten a altres països o que no estan disponibles en les altres plataformes. Amb tot, s'aconsegueix que la sèrie la puguin seguir moltes més persones que no pas si només s'emet a televisió un dia i ha una hora determinada.

A causa d'aquesta nova manera de fer, van sortir moltes pàgines web (amb les seves corresponents aplicacions per a mòbil) per ajudar als espectadors a portar un control sobre quines sèries i capítols han vist i quines els falta per veure, ja que moltes vegades es segueixen sèries per diferents plataformes i es fa difícil tenir-ne control.

Aquestes pàgines, són competència directe de la aplicació que es desenvoluparà en aquest TFG i estan explicades a continuació:

Tviso

Tviso va sorgir poc després de la desapareguda series.ly i es dedica a fer el mateix que series.ly però aquest cop no enllacen a cap pàgina per veure la sèrie de forma il·legal, es dediquen a enllaçar-ho a les plataformes legals actuals.

El seu sistema és bàsicament la de portar el control de la sèrie, l'usuari va marcant les sèries que segueix i els capítols que ha vist. Com a detall diferenciador està la de que el sistema t'avisa quan surt un capítol nou.

En aquesta plataforma la part "social" està en que pots deixar un comentari en cada capítol (hagis vist o no) i també valorar-les, i pots seguir a altres usuaris per veure'n la seva activitat.

Captures de pantalla i detalls del funcionament [aquí](#).

Pordede

Coneguda per ser la substituta a series.ly ja que té les mateixes funcionalitats exceptuant que no t'avisa de nous capítols i de que els comentaris només es poden fer a la sèrie en general.

Captures de pantalla i detalls del funcionament [aquí](#).

Touchvie

Aquesta es una aplicació una mica allunyada de les altres i de la pròpia aplicació d'aquest treball però té una component interessant i es que aquesta aplicació et permet tenir totes les dades del que està passant en un moment determinat del capítol. Quines sabates porta el protagonista, quina música sona, cotxe que passa i fins i tot anècdotes. A més et permet comprar la roba que porta el personatge o els detalls que surten, la música etc...

Més info [aquí](#).

Movistar+

Finalment, durant el passat Mobile World Congress movistar va presentar una funcionalitat per la seva televisió digital que seria la de poder comentar les sèries quan s'estan emeten en directe. Aquesta aplicació encara no està al mercat.

Aquesta és sens dubte és la competència directa amb l'aplicació d'aquest TFG ja que la possibilitat de comentar en directe és una de les funcionalitats diferenciadores que hi ha respecte la conferència. Faltarà examinar aquesta aplicació un cop surti per saber com funciona.

TFG: Creació d'una aplicació Android d'una xarxa social

Més info [aquí](#).

Per resumir, les pàgines actuals de referència per un seguidor de sèries totes segueixen el mateix patró: Un sistema per tenir el control dels capítols vistos i un lloc per deixar els comentaris de cada capítol així com també fer una valoració numèrica. També deixen els enllaços per poder veure el capítol, exceptuant el cas de Touchvie que no ho incorpora (és només app) i el cas de Movistar+ que estarà integrat al seu propi visualitzador.

L'aplicació d'aquest TFG serà per al sistema operatiu Android donada la seva alta penetració en el mercat dels smartphones i més encara en l'estat espanyol on és el sistema més utilitzat a molta distància del segon ("La obsesión española por Android I El Gadgetoblog I Blogs I elmundo.es," n.d.). S'ha optat per crear una aplicació nativa i no utilitzar frameworks com Phonegap ja que aquests normalment estan molt limitats i la seva experiència no és gens optima i pot fer que l'usuari tingui una mala experiència i deixi d'utilitzar l'aplicació.

Els llenguatges de programació utilitzats seran Java per l'Android i php+Json per la comunicació de l'aplicació amb la base de dades.

Metodologia

Aquest TFG tot i ser de caràcter individual, està compartit amb altres dos companys per a fer un projecte major. Això fa que per tal d'aconseguir que el resultat final sigui bo ens haguem de posar d'acord en molts aspectes i que tinguem un control del que fan els companys per a que tots seguim unes guies o pautes. Així doncs, en aquest TFG s'utilitza la metodologia àgil SCRUM combinada amb amb KANBAN.

SCRUM

SCRUM és un marc de treball per a equips que es basa en el manifest de la metodologia àgil i en el que es busca generar el mínim de documentació possible però procurant que tots els membres de l'equip sàpiguen exactament el que s'ha de fer i es van solucionant els problemes a mesura que van sortint tenint la opinió de tots els membres del grup.

Els esdeveniments que contempla SCRUM són els següents:

Sprints mensuals. Reunió de planificació d'sprints una vegada al mes.

Scrum diaris. Amb els membres del grup (TFG) per tal de portar un control sobre el que s'està fent i queda per fer.

Revisió de l'sprint. Per avaluar el resultat final.

Retrospectiva de l'sprint. Per si s'ha de repetir alguna part o passar el següent objectiu.

Equip SCRUM

Altrament, dins de la metodologia SCRUM existeixen diferents rols en els components del grup.

PROPIETARI DEL PRODUCTE

Es el professor que dirigeix el TFG. Ni que en un principi el propietari del producte es el responsable de gestionar la llista del producte, en aquest cas la responsabilitat serà compartida entre el professor i els estudiants. En aquesta llista s'expressen totes les característiques que tindrà la aplicació així com els detalls d'aquesta. Per tal de confeccionar aquesta llista s'utilitzen els programes com easybacklog per crear històries d'usuari que són una manera de posar en context les funcionalitats de l'aplicació.

EQUIP DE DESENVOLUPAMENT

Són els estudiants que treballen de forma autònoma cadascun referent al seu TFG i que s'organitzen de la forma que creuen més adient. En aquest aspecte, SCRUM no és molt exigent i reconeix per igual a tots els membres ja siguin desenvolupadors o artistes. Tothom treballa de forma autònoma en el seu TFG i ni l'SCRUM màster indica als membres de l'equip com han de treballar.

SCRUM MÀSTER

L'SCRUM màster és el responsable de procurar de que tots els membres de l'equip segueixin aquesta metodologia i de revisar constantment la feina feta per adaptar la feina i aconseguir acabar l'sprint mensual de forma correcte i sense problemes. Bàsicament es la persona encarregada de controlar i ajudar als altres membres del grup per tal d'aconseguir els objectius.

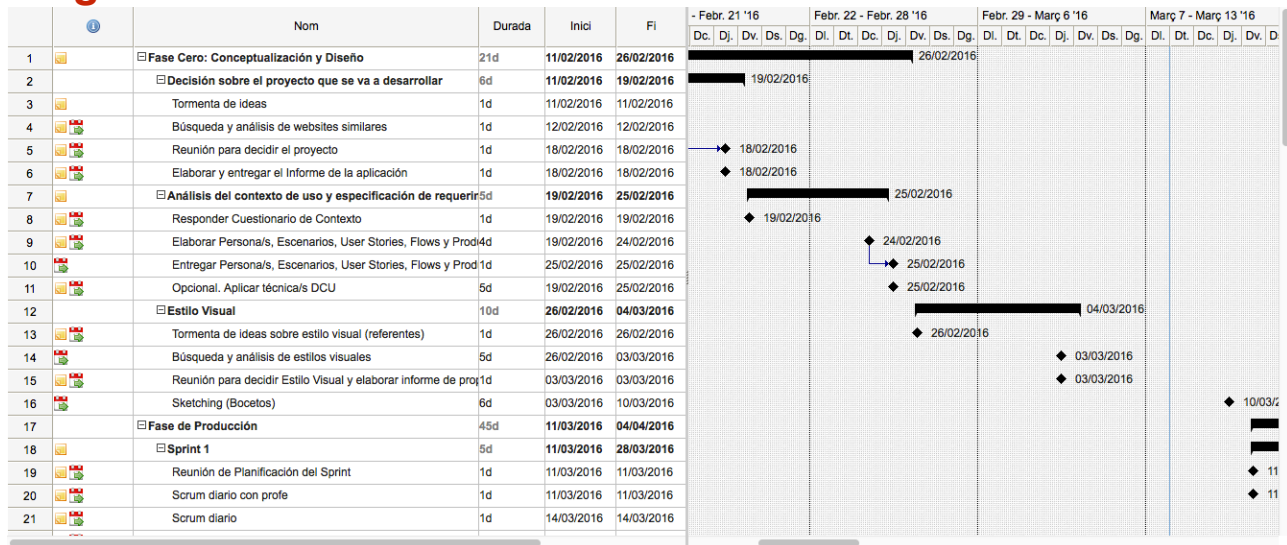
No hi ha delimitat un SCRUM màster fixe en aquest projecte, cada sprint mensual els alumnes es tornaran aquesta responsabilitat.

KANBAN

KANBAN es un mètode que proposa un tauler on apuntar totes les taques pendents en aquell moment separades per columnes depenent de si estan "pendents", en execució" o "acabades".

Planificació

Diagrama de Gant



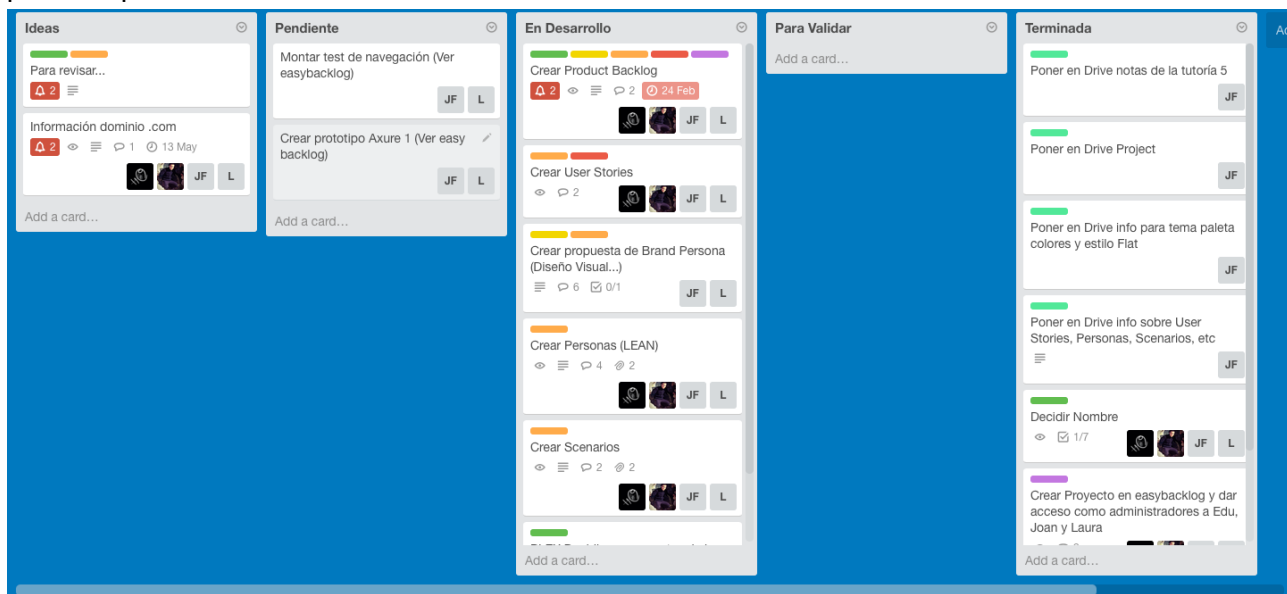
Utilitzant el Microsoft Project s'ha creat el diagrama de gant en el que es distingeixen dos fases del projecte.

La primera fase (Fase zero) és bàsicament la de conceptualització i el disseny. En aquesta fase es decidirà tot el referent a la xarxa social i per mitjà dels qüestionaris del context d'ús es veuran quins són els potencials usuaris i les tasques que poden fer, a part d'això també es crearan les user stories per tal de posar en context totes les funcionalitats de l'aplicació.

La segona fase és tot el referent a la producció. En aquesta part no s'han posat les tasques específiques (al diagrama de gant) sinó que s'han programat els sprints de la metodologia SCRUM, ja que serà en la reunió de l'sprint quan es dirà específicament les tasques a fer depenent de com estigui el projecte.

Trello

Aquest programa serveix per portar un control de tot el que estàs fent i queda per fer, es l'eina perfecte per utilitzar el mètode KANBAN.



Easybacklog

L'Easybacklog és una pàgina web que s'utilitza per crear les històries d'usuari que després s'utilitzen per tal de que tothom sàpiga perfectament quines funcionalitats té l'aplicació i en quin context es fa servir. Molt útil en les metodologies de treball àgil a on es necessita la mínima documentació possible i que tothom sàpiga en què es treballa.

<http://www.easybacklog.com/>

Aplicación móvil	ANM1	Como Usuario Quiero Registrarme Para poder Disponer de mi cuenta i poder acceder a la aplicación.	a) Test 2 de usabilidad con usuarios con ratio de eficacia igual o superior al 80% en tareas/s que impliquen identificar zonas de la interfaz b) Nombre, apellidos, dirección de correo electrónico y la contraseña son todos los campos requeridos c) Los campos adicionales incluyen la empresa y número de teléfono d) Dirección de correo electrónico debe ser único e) La contraseña debe pasar fuertes requisitos de contraseña f) Los nuevos registros se enviará un correo electrónico para validar la dirección de correo electrónico es válida	Igual que en la versión móvil	[edit]	0 €	0.0
	ANM2	Como Usuario Quiero Marcar un capítulo como visto. Para poder Llevar un control de las series y capítulos que a he visto	a) Poder marcar el capítulo en cuestión b) Sale un icono inequívoco de que está marcado		[edit]	0 €	0.0
	ANM3	Como Usuario Quiero Comentar un capítulo. Para poder Dar mi opinión.	a) Los campos de privacidad (para saber si el comentario es sólo para amigos o para todos) y el propio del comentario + Botón de enviar y que se guarde en el servidor		[edit]	0 €	0.0
	ANM4	Como Usuario Quiero Marcar un comentario como spam/inapropiado/etc. Para poder Para proteger a los usuarios que todavía no han visto el capítulo. O para que no vean el lenguaje inapropiado.	a) Poder marcar el comentario b) Mensaje de confirmación con los botones para aceptar o refusar c) Mensaje de comentario marcado correctamente d) Desaparece el mensaje		[edit]	0 €	0.0

Reunions

Aquest TFG, tal i com s'explica anteriorment, es fa en paral·lel amb altres dos TFG i es per això que, seguint la metodologia SCRUM, es fan reunions cada setmana amb el Product owner i reunions diàries (sobretot quan s'està en un sprint) amb els altres membres del grup per tal de portar un control de què s'està fent, què queda per fer i quines coses s'han de revisar per tal d'aconseguir un treball ràpid i fluid i que tothom pugui treballar sense esperes.

Skype

Es fa servir el programa Skype per tal de fer les reunions diàries amb els integrants del grup. Tot i això les reunions setmanals sí que son presencials.

<http://www.skype.com/>

Drive

Per tal de mantenir els arxius en un sol lloc i compartir els recursos necessaris es fa servir el Google Drive, al qual estan tots els integrants del grup inclòs el Product manager.

Whatsapp

Alternativament, els membres del grup (excepte el Product manager) utilitzen l'aplicació de whatsapp per els temes que requereixen urgència, discutir quan fer les reunions, problemes que poden sorgir, o temes relacionats amb el TFG.

Eines

Toggl

Aquesta aplicació bàsicament serveix per portar un control de les hores que dediques en un determinat treball. En aquest TFG s'utilitzarà per comptabilitzar les hores invertides, i un cop finalitzat es pot exportar un .pdf amb les hores finals especificades per dies/mesos i les tasques.

<http://www.toggl.com/>

Axure

El programa axure serveix per fer prototips d'aplicacions i pàgines web de forma ràpida i sense programar gaire, cosa que permet tenir un prototip ràpid per tal de provar-lo i solucionar qualsevol error abans no comenci la producció.

Android Studio

Android Studio és el programa per defecte per tal de poder compilar el programa perquè es pugui executar en un sistema Android, així que és un programa obligatori en aquest projecte.

Visual Studio Code

Programa de Microsoft que serveix per a programar de forma ràpida i intuïtiva. S'utilitzarà per a la creació dels documents php per el bescanvi de dades de servidor a Android.

Servidor

Es necessari un servidor per tal de crear la base de dades i els arxius PHP necessaris per tal de que l'aplicació es pugui comunicar amb la base de dades. No serà necessari un servidor diferent a la configuració més comuna, Apache (HTML+PHP+MySQL).

Pressupost

Per tal de fer el pressupost partim de la base de que som una petita empresa que ja ha fet una inversió prèvia de material i, llavors, només paga amortitzacions d'aquesta.

Part fixe	Quantitat	Preu	Amortització
Alquiler local - 40m2	1	200,00 €	200,00 €
Ordenadores	3	3.150,00 €	65,63 €
Internet	1	40,00 €	40,00 €
Tablet	1	250,00 €	10,42 €
Smartphone	3	800,00 €	16,67 €
Licencia software	1	60,49 €	60,49 €
Android developer fee	1	25,00 €	2,08 €
Dominio	1	1,00 €	0,08 €
Servidor	1	2,99 €	2,99 €
		TOTAL:	398,36 €
Part variable	Quantitat	Preu	
Tracker	1	por determinar	
Alquiler laboratorio HCI	1	por determinar	

398,36€/mes son les despeses mensuals que l'empresa ha de fer front. A part d'aquestes despeses s'hi ha d'afegir el preu/hora de les hores treballades. ~~S'estima el projecte en 3 mesos així que el total són: 1195,08€~~ El projecte ha durat 6 mesos així que el cost total es de 2390,16€.

Per tal de calcular el salari dels treballadors, s'utilitzarà un sistema de punts basat en el programari easybacklog en el qual cada tasca té uns punts associats seguint la successió de Fibonacci i cada punt equival a un cost monetari.

Seguint el model que utilitzen a easybacklog en el que 2,7 dies equivalen a 8 punts i tenint en conte que es calcula un treball diari de 3h, ens surt que cada punt equival a quasi una hora.

Com a idea, s'ha pensat que una manera d'aconseguir ingressos amb l'aplicació sigui la de vendre a persones interessades com podrien ser guionistes o productores, un anàlisis de les dades que tindrem de gustos i opinions de la sèrie. Aquesta idea encara està en fase de conceptualització.

Màrqueting

En aquest apartat se'n poden distingir dues parts, una destinada al màrqueting de l'aplicació i una altre més destinada a la marca personal.

Aplicació (TFG)

Pel que fa al màrqueting referent a l'aplicació, es distingeixen a aquelles persones de més de 13 anys a les que els agraden les sèries i poder-les comentar amb altres usuaris i amics, i als que la tecnologia no els suposa un obstacle sinó un aliat.

Un cop identificats els usuaris potencials i un cop desenvolupada l'aplicació, la forma de donar-se a conèixer, i que pot resultar més efectiva, serà la del boca-orella, ja que en el fons els usuaris es comunicaran amb els seus amics i es una forma bastant ràpida i senzilla perquè la gent provi i s'enganxi a l'aplicació.

Amb el boca-orella també entra el tema d'utilitzar les xarxes socials, per exemple fent que els usuaris comparteixin l'aplicació per allà perquè els seus contactes s'hi apuntin.

Paral·lelament també es contempla la realització d'una nota de premsa i d'una vídeo presentació que de totes maneres s'ha de fer igual per a la presentació del projecte.

Un cop alliberada la versió final de l'aplicació, s'utilitzarà el framework fabric.io per a tenir analítiques de la utilització i informació d'errors i així portar un control de cara a millorar l'experiència d'usuari i detectar qualsevol problema abans que l'usuari abandoni l'aplicació.

Marca personal

Pel que fa al màrqueting referent a la part de marca personal, paral·lelament al desenvolupament del TFG, tinc previst fer les següents accions per tal d'aconseguir que acabat el projecte de final de grau pugui tenir una marca personal sòlida.

Primerament serà la d'estar en diferents xarxes socials per tal de que la gent em pugui trobar fàcilment, i les xarxes que utilitzaré seran LinkedIn, Twitter i Google+ (aquesta última per un tema de seo al cercador de Google). Facebook no entra dins dels plans ja que la deixo per temes més personals, a l'igual que instagram ja que no crearé material gràfic. Es important que en totes les xarxes socials mantingui la foto de perfil (i/o les de portada) per tal de donar solidesa a la marca.

Alternativament a les xarxes socials, també utilitzaré una web personal a on mostrar els treballs realitzats, en definitiva com un cv però en web (amb disseny i personalitat pròpia). També està pensada la creació d'un bloc al mateix web i en el que en un principi podria posar diferents posts referents a aquest projecte que segurament dona per moltes entrades.

Finalment, i no menys important, també estaria la tècnica utilitzada per a l'aplicació que seria el boca-orella, ja que ni que tradicional és la que més funciona.

Desenvolupament

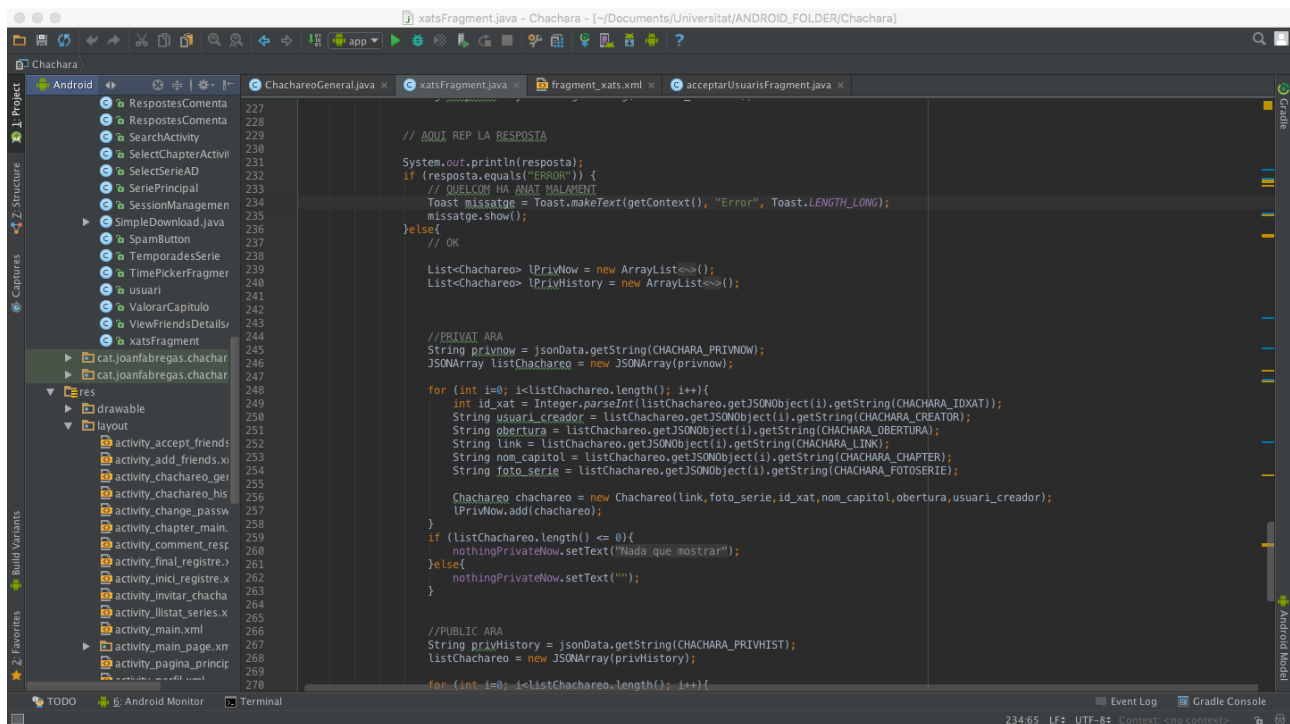
Android

Per què Android ?

Tal i com ja he esmentat en anterioritat les raons de programar l'aplicació amb Android són primerament per la voluntat que tenia l'any passat de cursar l'assignatura d'Android que ofería el centre i que aquest any no vaig poder fer i també, tal i com mostra l'estudi esmentat amb anterioritat, Android es el sistema operatiu mòbil que més s'utilitza a l'estat espanyol i a la majoria del món.

Android Studio

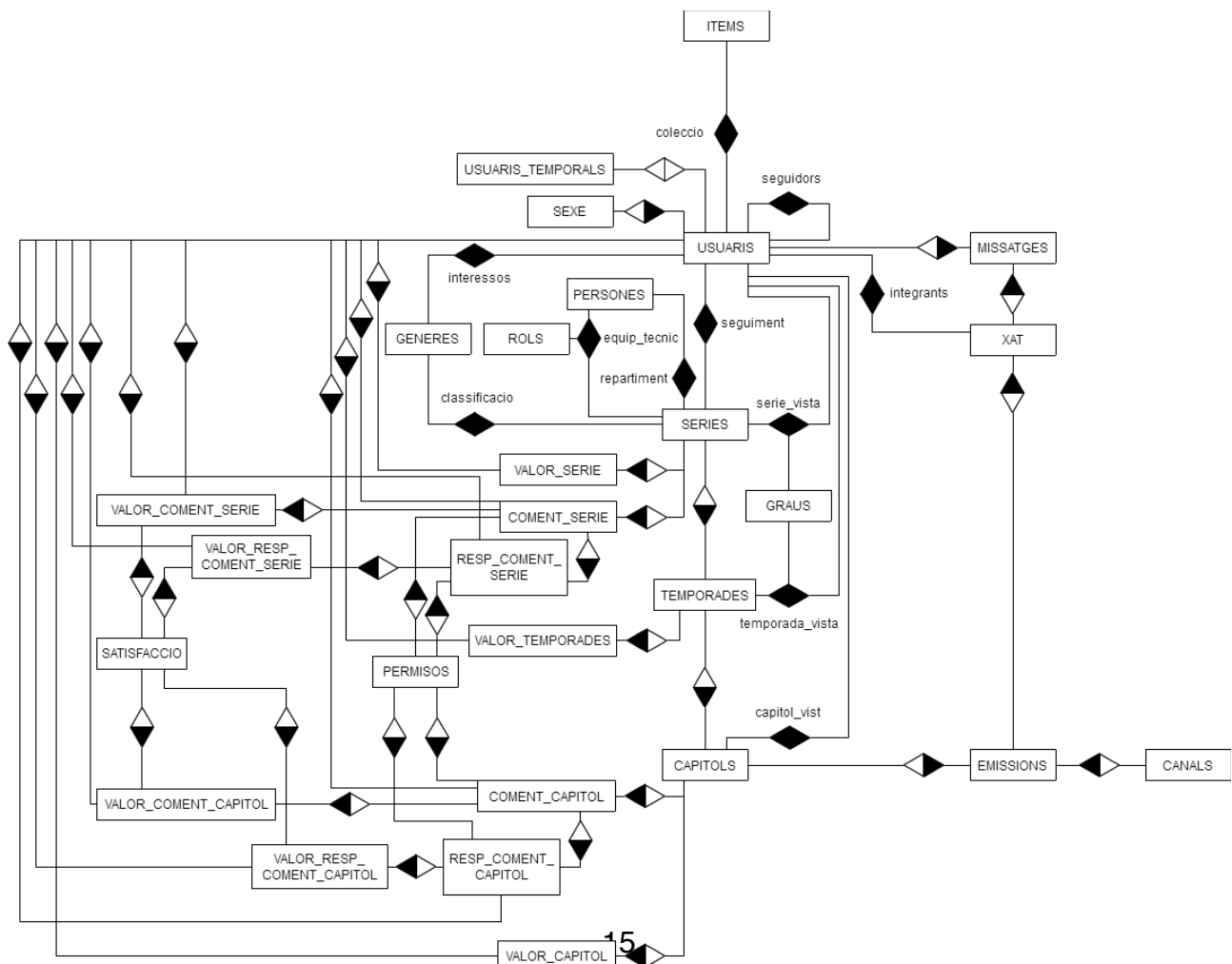
Per tal de programar en Android es d'ús necessari el programa Android Studio de la pròpia Google. Aquest programa que es basa en un programa existent (intelliJ) per programar en Java, està desenvolupat estrictament per programar per a Android i es per això que afegeix moltes millores i frameworks per tal d'ajudar al programador.



INSTANTÀNIA DEL PROGRAMA ANDROID STUDIO AMB CODI DE L'APLICACIÓ CHÁCHARA

```
{
  "response": "OK",
  "series": [{
    "id_serie": "1",
    "nom": "Stranger Things",
    "descripcio": "lorem ipsum dolor sit amet",
    "imatge": "http:\\\\citmalumnes.upc.es\\~G11IDI\\img_series\\StrangerThings.jpg",
    "isFollowing": true
  }, {
    "id_serie": "2",
    "nom": "The Big Bang Theory",
    "descripcio": "lorem ipsum dolor sit amet",
    "imatge": "http:\\\\citmalumnes.upc.es\\~G11IDI\\img_series\\bbt.jpg",
    "isFollowing": false
  }]
}
```

Base de dades Estructura



BBDD i Android

Com es fa?

Com que la informació de Chàchara es troba allotjada al servidor, es necessita d'un diàleg entre la base de dades del servidor i l'aplicació Android d'aquest treball. Per dur a terme aquest diàleg, s'utilitzen documents php allotjats al servidor els quals son cridats per l'aplicació i executats al servidor i el qual retorna les dades d'una manera que la aplicació pugui utilitzar-les. Android pot llegir i utilitzar la informació de servidor si les rep amb llenguatge XML o JSON. En aquest treball s'utilitza el JSON per la seva senzillesa.

Comunicació JSON - JAVA

Per posar un exemple de la comunicació entre servidor i aplicació es detalla el funcionament del document **appLlistatSeries.php** encarregat de mostrar una llista amb informació de les series. Aquesta es la resposta que el document retorna a l'aplicació Android:

Aquesta informació es llegeix a l'aplicació com a objecte JSON i es fan les accions oportunes.

```
final String CHACHARA_RESPONSE = "response";
final String CHACHARA_NOMSERIE = "nom";
final String CHACHARA_DESCSERIE = "descricio";
final String CHACHARA_IMGSERIE = "image";
final String CHACHARA_ISFOLLOWING = "isFollowing";
final String CHACHARA_LLISTASERIES = "series";
final String CHACHARA_IDSIE = "id_serie";

try{
    JSONObject jsonData = new JSONObject(getmData());
    String resposta = jsonData.getString(CHACHARA_RESPONSE);

    // AQUI REP LA RESPOSTA

    System.out.println(resposta);
    if (resposta.equals("ERROR")) {
        // QUEL COM HA ANAT MALAMENT
        Toast missatge = Toast.makeText(getApplicationContext(), "Error", Toast.LENGTH_LONG);
        missatge.show();
    }else{
        // OK

        String ser = jsonData.getString(CHACHARA_LLISTASERIES);
        JSONArray series = new JSONArray(ser);

        List<ChacharaSeries> chacharaSeriesList = new ArrayList<>();

        for(int i=0 ; i<series.length() ; i++){
            String nomSerie = series.getJSONObject(i).getString(CHACHARA_NOMSERIE);
            String descSerie = series.getJSONObject(i).getString(CHACHARA_DESCSERIE);
            String imgSerie = series.getJSONObject(i).getString(CHACHARA_IMGSERIE);
            String idSerie = series.getJSONObject(i).getString(CHACHARA_IDSIE);
            String isFollowingSerie = series.getJSONObject(i).getString(CHACHARA_ISFOLLOWING);

            ChacharaSeries serie_sola = new ChacharaSeries(descSerie,Integer.parseInt(idSerie),Boolean.parseBoolean(isFollowingSerie),nomSerie,imgSerie);
            chacharaSeriesList.add(serie_sola);
        }

        mChacharaSeriesAdapter.loadNewData(chacharaSeriesList);
    }
}
```

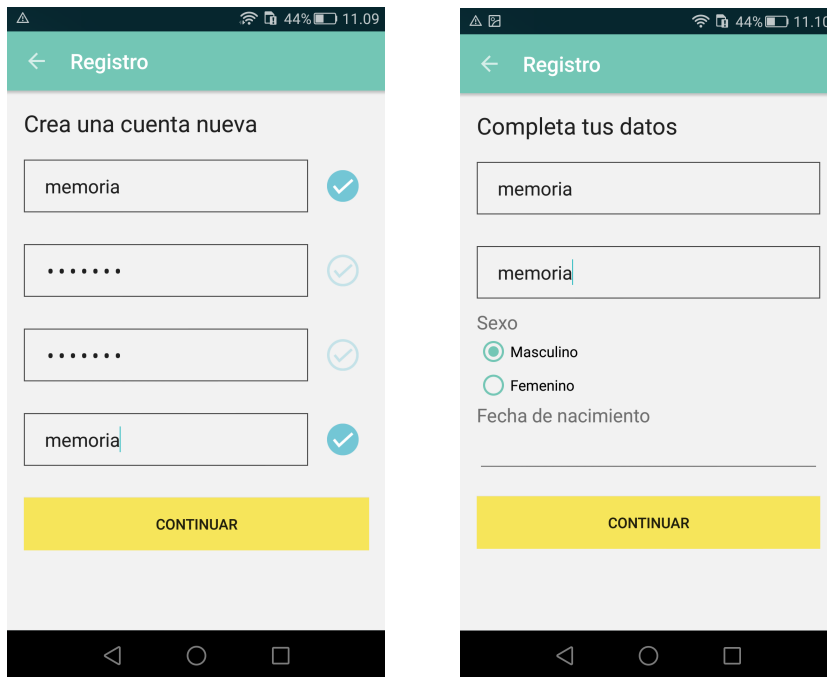
En aquest cas el codi llegeix la informació i crea una llista amb la informació de les series que ha retornat el servidor. Posteriorment s'utilitza aquesta llista per mostrar les series a l'usuari.

Programació de l'aplicació

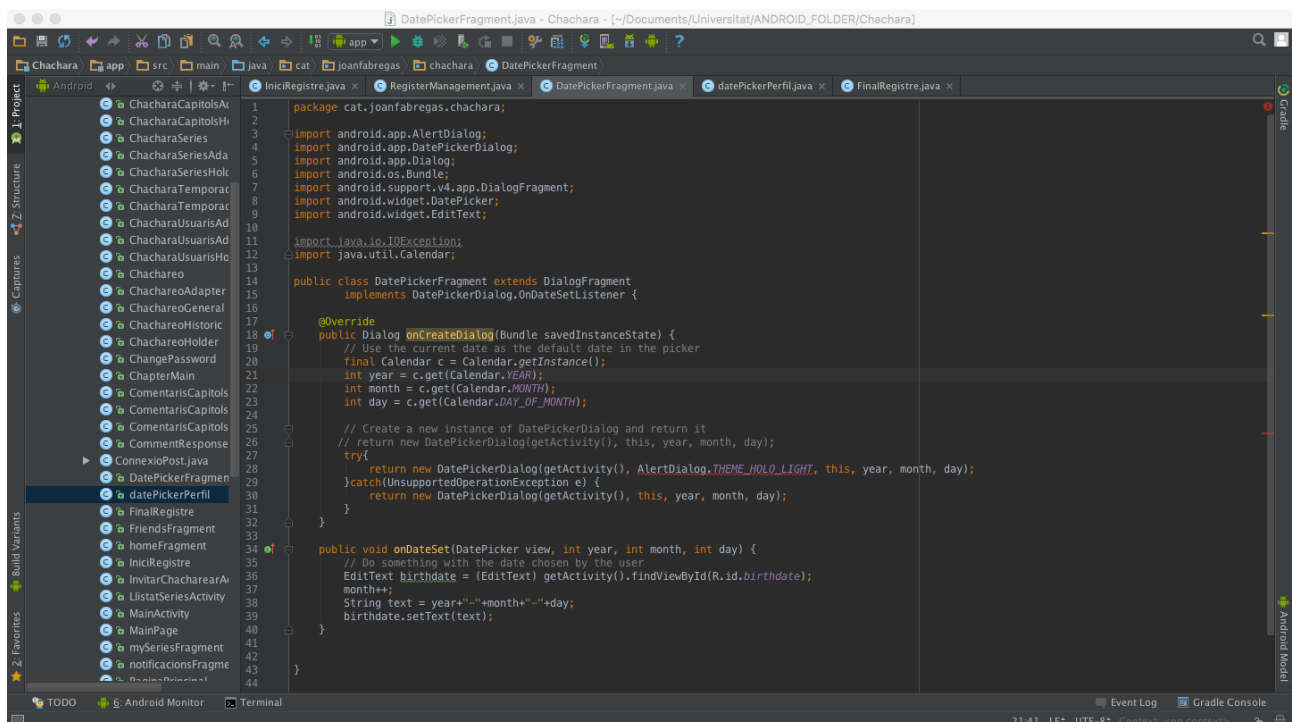
Registre/Inici Sessió + Perfil

Registre

Aquesta part consta de dos pàgines en les quals primer l'usuari ha d'incorporar les dades que farà servir per a entrar a l'aplicació i després les seves dades personals.

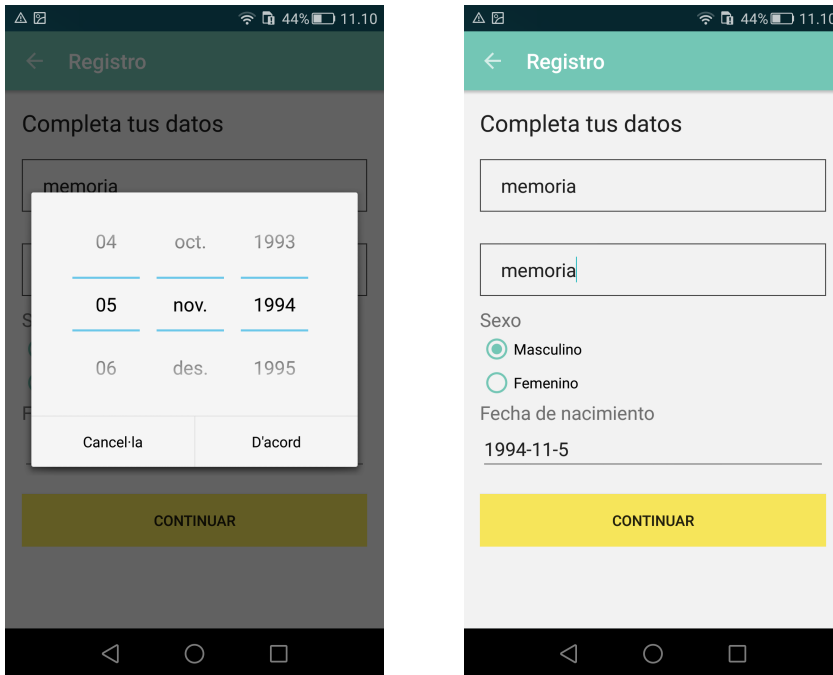


En aquest apartat s'implementa la funcionalitat **DatePickerDialog** de l'Android per tal de que apareixés un diàleg i que l'usuari pogués afegir la seva data de naixement de forma més fàcil.

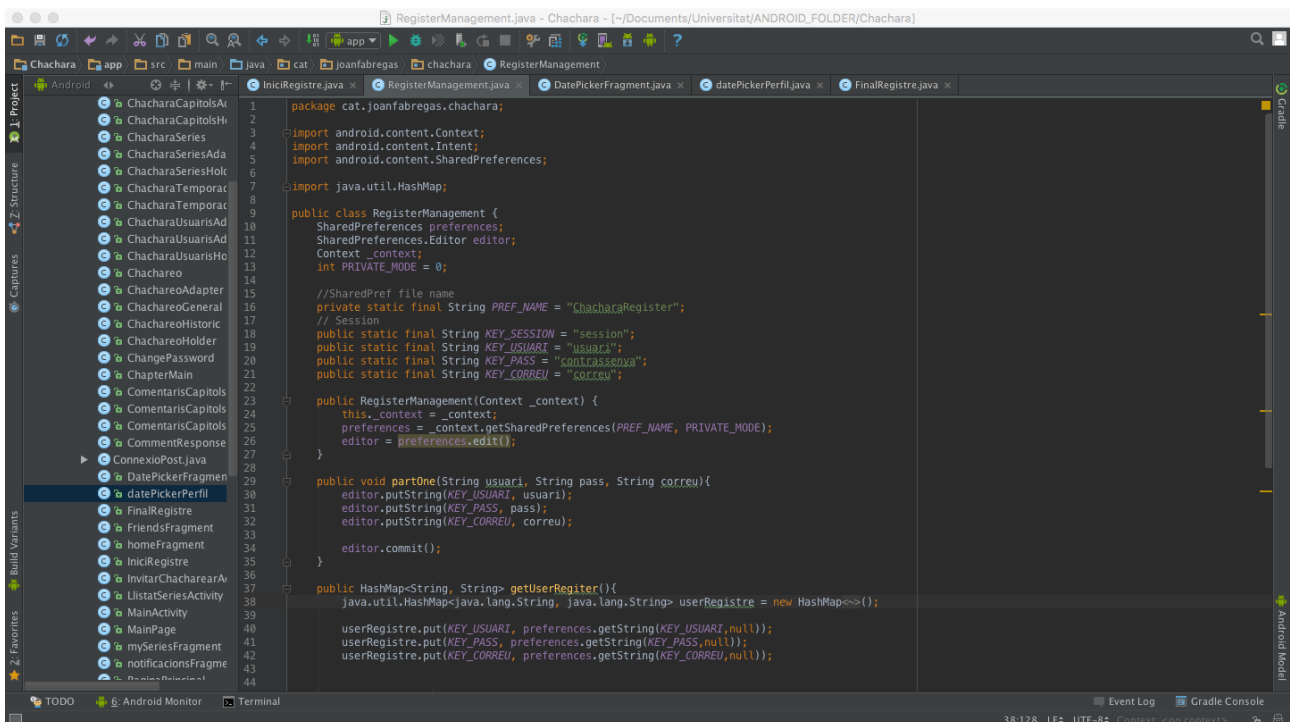


TFG: Creació d'una aplicació Android d'una xarxa social

CODI DATEPICKERDIALOG



En acabar aquesta informació es registra al servidor per tal de que quedi contància del registre. Per tal de desar les dades de l'usuari perquè li sigui més ràpid i fàcil entrar a l'aplicació es fa ús de la funcionalitat de preferències compartides (**SharedPreferences**) de l'Android. També s'utilitzen per emmagatzemar les dades prèvies al registre.

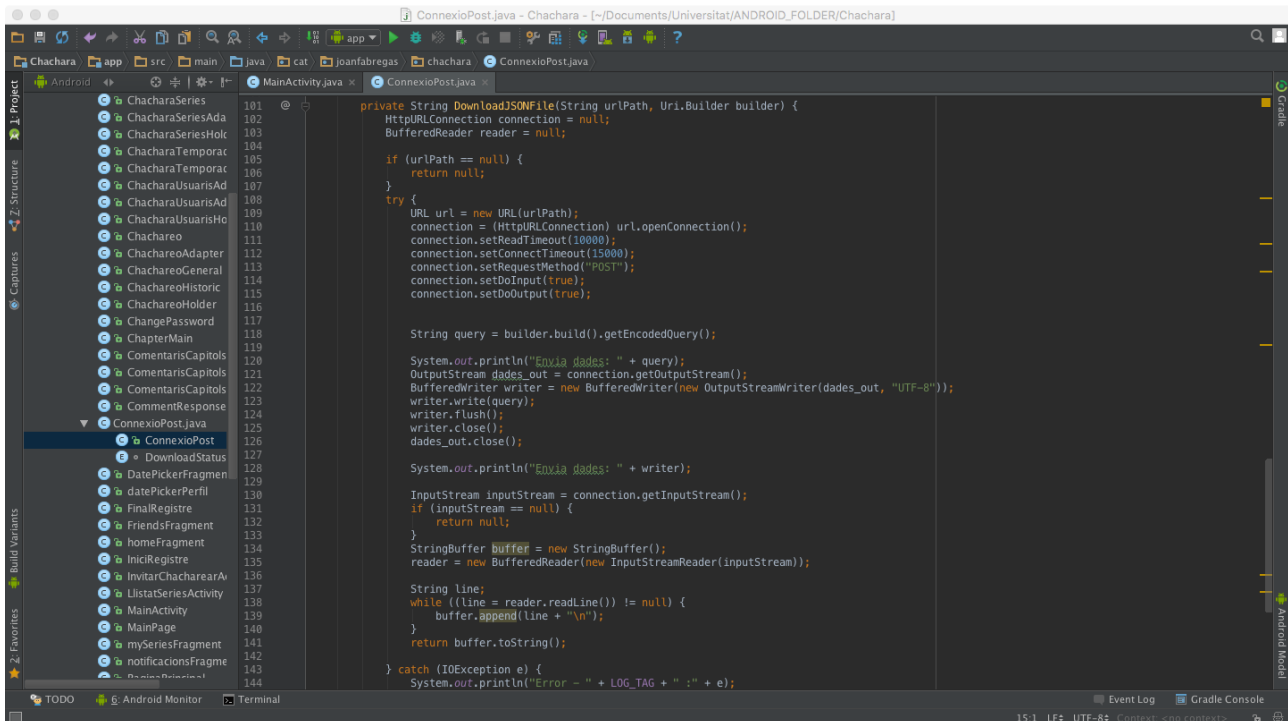


SHAREDREFERENCES DETALL DEL CODI

TFG: Creació d'una aplicació Android d'una xarxa social

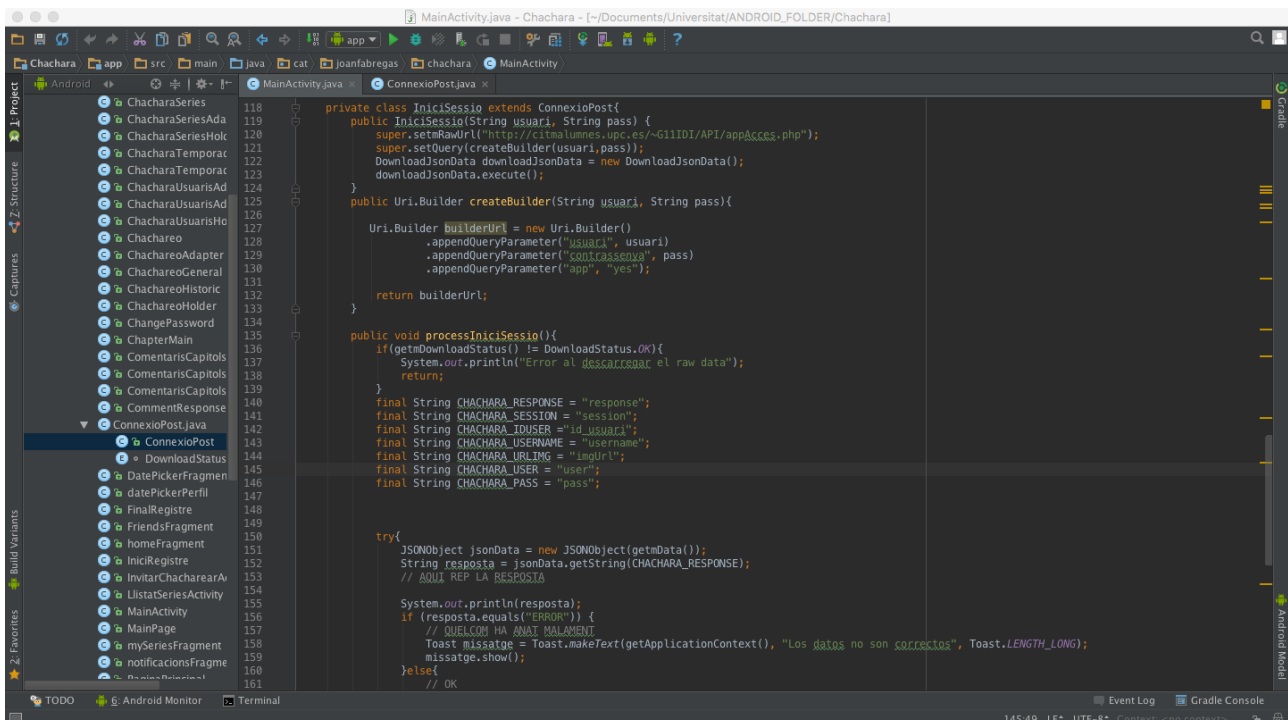
Per tal d'enviar les dades o de rebre-les, es fa servir la funcionalitat **HttpURLConnection**. Aquest procediment es fa servir en tota l'aplicació per l'enviament de peticions HTTP i serveix per enviar i rebre les dades.

L'arxiu encarregat d'utilitzar aquesta funcionalitat es ConnexioPost.java, que es el que utilitzen tots els altres arxius on hi ha connexió amb el servidor.



EXEMPLE DE PART DEL CODI DE CONNEXIOPOST.JAVA

En aquest hi ha detallat tot el procediment per enviar i rebre les dades, però és en els altres documents java a on s'expandeix aquesta classe i se l'hi incorpora la informació sobre quines dades enviar, a on enviar-les i com processar-les.



MOSTRA DEL CODI QUE EXPANDEIX LA CLASSE I HI AFEGEIX LA INFORMACIÓ NECESSÀRIA I COM PROCESSAR-LA

Inici sessió

Per iniciar sessió l'usuari pot afegir les seves dades o si s'acaba de registrar ja estaràn incorporades ja que com s'ha comentat es desen a SharedPreferences. Les dades que s'hi emmagatzemen están encriptades així que es mantenen segures.



Com es pot observar es fa servir el HashMap per poder aconseguir la informació:

```
// Session class instance
session = new SessionManagement(getApplicationContext());
// get user data from session
HashMap<String, String> user = session.getUserDetails();
session_id = user.get(SessionManagement.KEY_SESSION);
```

EN AQUEST CAS ES RETORNA LA INFORMACIÓ REFERENT A LA SESSIÓ CLIENT-SERVIDOR

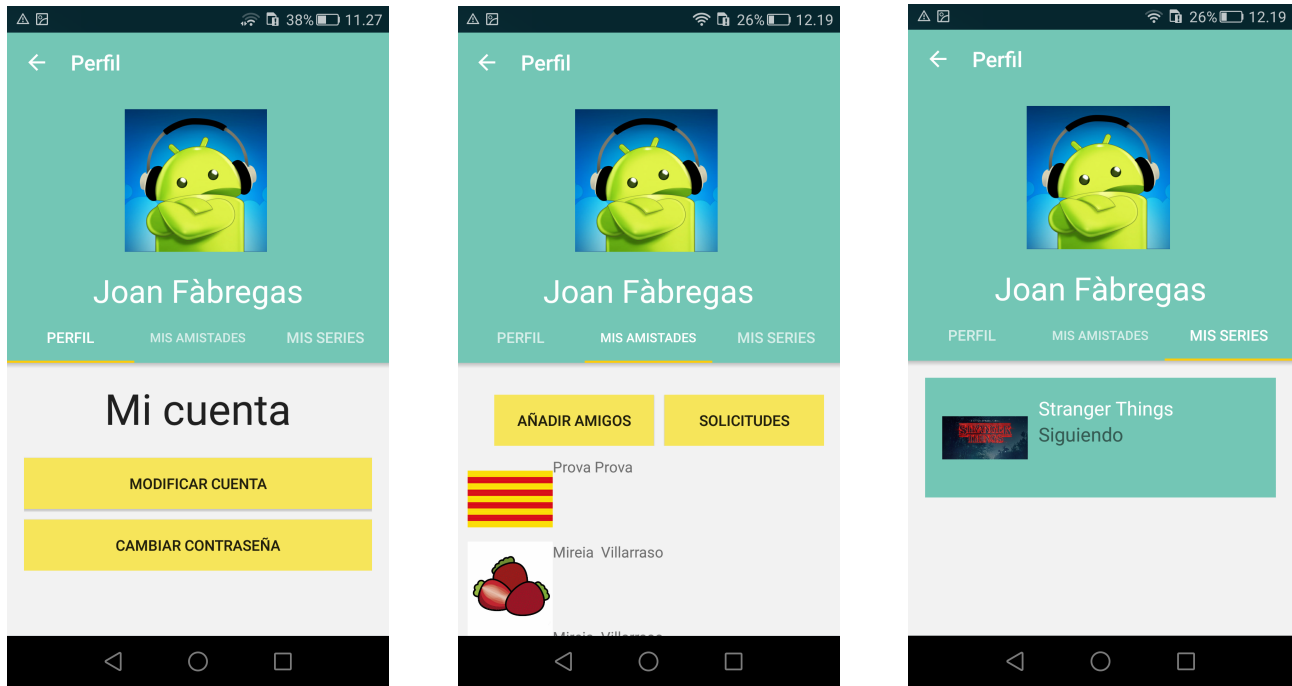
Com es pot observar, en les preferències es desa la informació referent a la sessió (codi que serveix al servidor per identificar a la màquina amb la qual interactua). Aquest codi es va haver de desar ja que si no enviéssim aquest codi quan fem una petició http al servidor es pensaria que som un usuari diferent i no ens reconeixaria. Per això en tots els php del servidor mirem si se l'hi envia la sessió per dir-li que som els mateixos.

```
//sessio APP
if (isset($_POST['session']) && $_POST['session']) {
    session_id($_POST['session']);
}
/*
```

CODI DELS DOCUMENTS PHP REFERENTS A LA SESSIÓ

Perfil

En el perfil a part de trobar la informació relacionada amb els usuaris, es troben també les seves amistats i les sèries que està seguint. Per tal d'incorporar tota aquesta informació i seguint el disseny proposat, s'utilitza una nova funcionalitat. En aquest cas es tracta de incorporar "tabs", de manera que l'usuari es mou horitzontalment per les pantalles.



Cada pantalla s'anomena Fragment, que té el seu corresponent fitxer Java i XML pel disseny com si es tractés d'una pantalla normal, i la pantalla inicial s'ha d'incorporar la funcionalitat `FragmentManager` per tal de que funcioni.

Els fragments tenen un codi molt semblant a les pàgines normals, però les seves classes extenen de la classe `Fragment` i per tant hi ha alguna diferència. La principal es que mentre que les normals tot el tema de linkar els botons i texts al codi es fa a la funció `onCreate()`, aquí es fa a la funció `onCreateView()` ja que la "view" es crea a posteriori que s'executa l'script.

Exemple del codi de la pàgina a on hi ha els fragments i com s'executen i administren:

```
public void Perfil(){
    Intent intent = new Intent(this, perfilActivity.class);
    startActivity(intent);
}
private void setupViewPager(ViewPager viewPager){
    ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());

    adapter.addFragment(new perfilFragment(), "Perfil");
    adapter.addFragment(new FriendsFragment(), "Mis Amistades");
    adapter.addFragment(new mySeriesFragment(), "Mis Series");
    viewPager.setAdapter(adapter);
}
class ViewPagerAdapter extends FragmentPagerAdapter {
    private final List<Fragment> mFragmentList = new ArrayList<>();
    private final List<String> mFragmentTitleList = new ArrayList<>();

    public ViewPagerAdapter(FragmentManager manager) {
        super(manager);
    }

    @Override
    public Fragment getItem(int position) {
        return mFragmentList.get(position);
    }

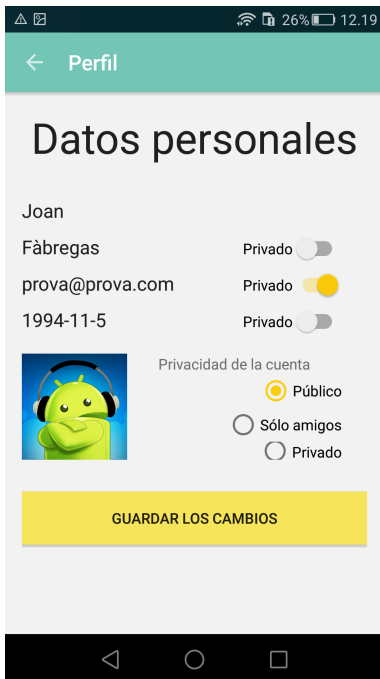
    @Override
    public int getCount() {
        return mFragmentList.size();
    }

    public void addFragment(Fragment fragment, String title) {
        mFragmentList.add(fragment);
        mFragmentTitleList.add(title);
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return mFragmentTitleList.get(position);
    }
}
```

CODI DE LA PANTALLA PRINCIPAL QUE HI HA ELS FRAGMENTS

A l'apartat de Perfil també hi ha dos implementacions més de Android referents més aviat al disseny. Que serien els botons estil on/off i els "radio buttons" per seleccionar una opció.

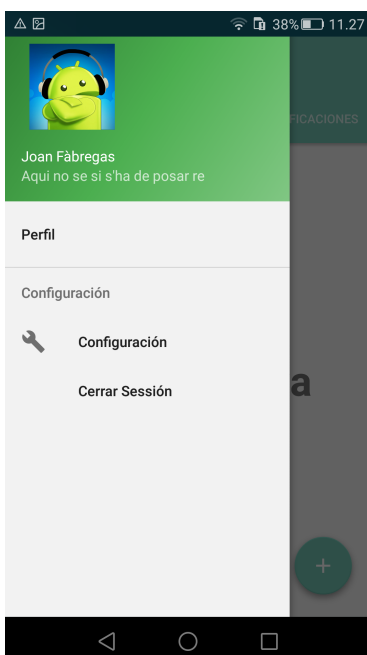


Series + Pàgina principal

Pàgina Principal

A la pàgina principal de l'aplicació es segueix el mateix procediment que a perfil i es compona de 3 fragments. La configuració és la mateixa que a Perfil però en aquest cas les funcions que fan els fragments es diferent.

A part dels fragments, aquesta part té dos funcionalitats importants. La primera es el Menú, està format amb la implementació **DrawerLayout**, que s'encarrega d'afegir un *layout* diferent al que ja té la pàgina que serà el menú i que sortirà al prémer el botó de menú des de l'esquerra. boto flotant + hamburguesa.




```

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);

    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.perfil) {
        Intent perfil = new Intent(this, MainPage.class);
        startActivity(perfil);
    } else if (id == R.id.nav_manage) {
        //Configuració
    } else if (id == R.id.nav_logout) {
        session.logoutUser();
    }

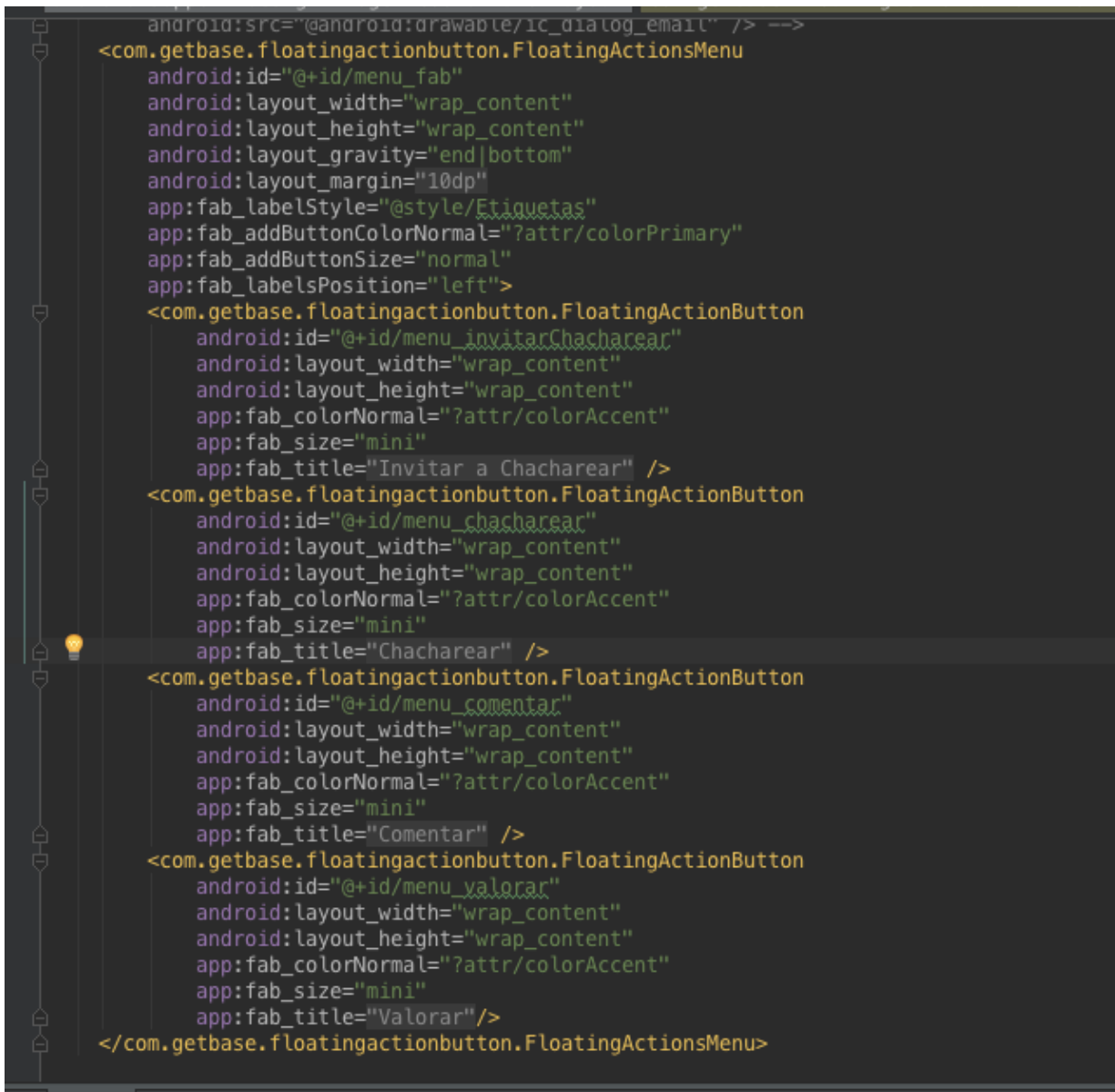
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

```

PROGRAMACIÓ DEL MENÚ

En la imatge es pot observar el funcionament del *DrawerLayout* i de les funcions que fa el programa depenent de a què faci click l'usuari.

En aquesta pàgina també apareix un botó flotant d'accés directe en que l'usuari pot accedir ràpidament a les funcionalitats bàsiques. Al contrari que la majoria d'implementacions, aquest botó flotant té suport natiu i no cal cap implementació en les classes java especials per tal de que funcioni. Funciona afegint el codi xml als documents que fan referència a l'apartat gràfic.



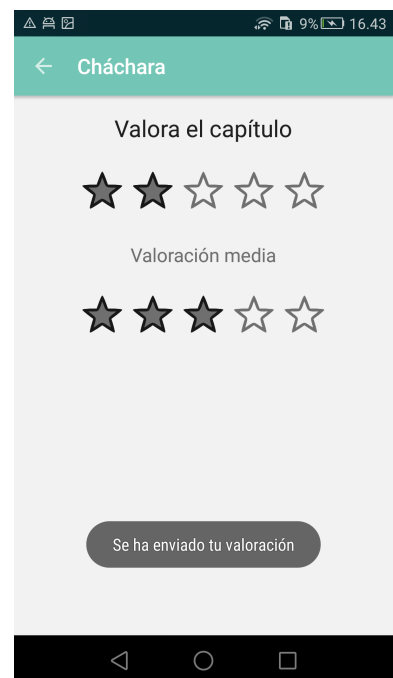
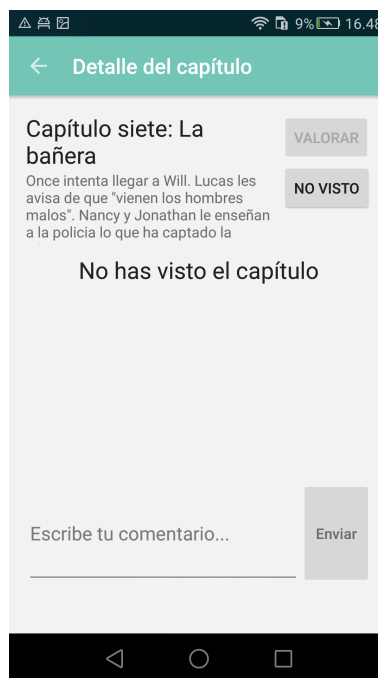
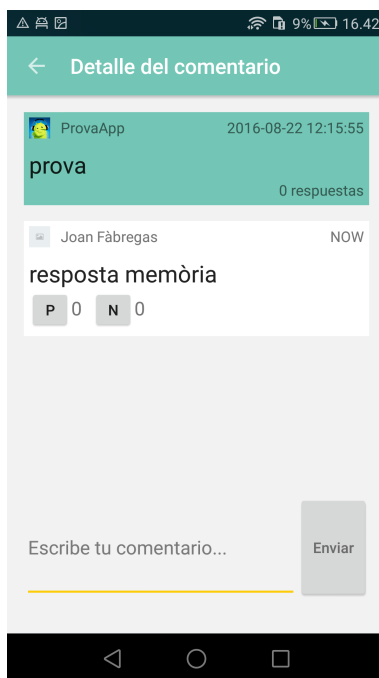
XML DEL BOTÓ FLOTANT D'ACCÉS DIRECTE

Sèries

En aquest apartat l'usuari es capaç de buscar una sèrie o seleccionar-la d'una llista, veure informació d'aquesta, seguir-la i veure'n els capítols que té. Seleccionant la temporada i el capítol l'usuari arriba a la informació del capítol on hi ha els comentaris i les valoracions d'altres usuaris. Si l'usuari no ha marcat el capítol com a vist, però, no podrà veure cap comentari ni valorar-lo.

A continuació les imatges referents a la app:

TFG: Creació d'una aplicació Android d'una xarxa social



En aquest apartat s'han d'utilitzar llistes d'informació dinàmica que s'ha de mostrar a l'usuari i ha resultat ser el més difícil del treball. Per posar un exemple s'explica el procés utilitzant la pantalla de selecció de la sèrie.

Primerament s'ha de crear un document php que et retorni les dades de les sèries i cridar la pàgina utilitzant el sistema explicat anteriorment. A continuació i abans de llegir les dades del servidor primer s'ha de crear un fitxer xml que fa referència a com es veurà a l'aplicació. (S'explica el procés més endavant) i llavors es crea un document java fent referència als objectes gràfics.

```
public class ChacharaSeriesHolder extends RecyclerView.ViewHolder {
    protected ImageView imageSerie;
    protected TextView nameSerie;
    protected TextView followSerie;

    public ChacharaSeriesHolder(View itemView) {
        super(itemView);
        this.followSerie = (TextView) itemView.findViewById(R.id.followSerie);
        this.imageSerie = (ImageView) itemView.findViewById(R.id.imageSerie);
        this.nameSerie = (TextView) itemView.findViewById(R.id.nameSerie);
    }
}
```

CODI QUE ENLLAÇA EL GRAFISME AMB EL CODI

Després, s'ha de crear un objecte per cada sèrie que tinguem. Per a fer-ho es crea una classe en què es passen totes les dades. En aquesta classe s'incorporen funcions per a després poder accedir a la informació.

```
public class ChacharaSeries {
    private int mId_serie;
    private String mNom;
    private String mDescripcio;
    private String mUrlImage;
    private boolean mIsFollowing;

    public ChacharaSeries(String descripcio, int id_serie, boolean isFollowing, String nom, String urlImage) {
        mDescripcio = descripcio;
        mId_serie = id_serie;
        mIsFollowing = isFollowing;
        mNom = nom;
        mUrlImage = urlImage;
    }

    public String getDescripcio() {
        return mDescripcio;
    }

    public int getId_serie() {
        return mId_serie;
    }

    public boolean isFollowing() {
        return mIsFollowing;
    }

    public String getNom() {
        return mNom;
    }

    public String getUrlImage() {
        return mUrlImage;
    }
}
```

CLASSE CHACHARASERIES.JAVA

Després, utilitzant el framework *RecyclerView* de l'Android es crea una classe que serà l'encarregada de mostrar totes les series. Aquesta classe està enllaçada amb el primer codi i s'hi afegeix un contenidor per a posar-hi tots els objectes sèrie que es puguin crear.

```

public class ChacharaSeriesAdapter extends RecyclerView.Adapter<ChacharaSeriesHolder>{
    protected List<ChacharaSeries> mChacharaSeries;
    private Context mContext;

    public ChacharaSeriesAdapter(List<ChacharaSeries> chacharaSeries, Context context) {
        mChacharaSeries = chacharaSeries;
        mContext = context;
    }

    @Override
    public ChacharaSeriesHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.series_list,null);
        ChacharaSeriesHolder chacharaSeriesHolder = new ChacharaSeriesHolder(view);
        return chacharaSeriesHolder;
    }

    @Override
    public void onBindViewHolder(ChacharaSeriesHolder holder, int position) {
        ChacharaSeries chacharaSeries = mChacharaSeries.get(position);
        if(chacharaSeries.isFollowing()){
            holder.followSerie.setText("Siguiendo");
        }else{
            holder.followSerie.setText("");
        }
        holder.nameSerie.setText(chacharaSeries.getNom());
        Picasso.with(mContext).load(chacharaSeries.getUrlImage())
            .error(R.drawable.placeholder)
            .into(holder.imageSerie);
    }

    @Override
    public int getItemCount() {
        return (null != mChacharaSeries ? mChacharaSeries.size() : 0);
    }

    public ChacharaSeries getSerie(int position){
        return (null != mChacharaSeries ? mChacharaSeries.get(position) : null);
    }

    public void loadNewData(List<ChacharaSeries> newSerie){
        mChacharaSeries = newSerie;
        notifyDataSetChanged();
    }
}

```

CLASSE CHACHARASERIESADAPTER.JAVA

Un cop tenim aquests codis apunts, s'afegeix a la part gràfica de l'aplicació el "view" per tal de mostrar-li a on s'ha d'afegir tot el contingut dinàmic.

```

<view
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="android.support.v7.widget.RecyclerView"
    android:id="@+id/recycler_viewSeries"
    app:layoutManager="android.support.v7.widget.LinearLayoutManager"/>

```

XML DEL "VIEW" QUE DETERMINA A ON S'HA DE VEURE

En el codi de la pàgina s'ha d'afegir aquest "Adaptador" que s'acaba de crear i s'enllaça amb el "view" i llavors només cal descarregar la informació tal i crear els objectes sèrie necessaris i aquests s'aniran mostrant a l'aplicació.

```
final String CHACHARA_RESPONSE = "response";
final String CHACHARA_NOMSERIE = "nom";
final String CHACHARA_DESCSERIE = "descripcio";
final String CHACHARA_IMGSERIE = "imatge";
final String CHACHARA_ISFOLLOWING = "isFollowing";
final String CHACHARA_LLISTASERIES = "series";
final String CHACHARA_IDSERIE = "id_serie";

try{
    JSONObject jsonData = new JSONObject(getmData());
    String resposta = jsonData.getString(CHACHARA_RESPONSE);

    // AQUI REP LA RESPOSTA

    System.out.println(resposta);
    if (resposta.equals("ERROR")) {
        // QUEL'COM HA ANAT MALAMENT
        Toast missatge = Toast.makeText(getApplicationContext(), "Error", Toast.LENGTH_LONG);
        missatge.show();
    }else{
        // OK

        String ser = jsonData.getString(CHACHARA_LLISTASERIES);
        JSONArray series = new JSONArray(ser);

        List<ChacharaSeries> chacharaSeriesList = new ArrayList<>();

        for(int i=0 ; i<series.length() ; i++){
            String nomSerie = series.getJSONObject(i).getString(CHACHARA_NOMSERIE);
            String descSerie = series.getJSONObject(i).getString(CHACHARA_DESCSERIE);
            String imgSerie = series.getJSONObject(i).getString(CHACHARA_IMGSERIE);
            String idSerie = series.getJSONObject(i).getString(CHACHARA_IDSERIE);
            String isFollowingSerie = series.getJSONObject(i).getString(CHACHARA_ISFOLLOWING);

            ChacharaSeries serie_sola = new ChacharaSeries(descSerie,Integer.parseInt(idSerie),Boolean.parseBoolean(isFollowingSerie),nomSerie,imgSerie);
            chacharaSeriesList.add(serie_sola);
        }

        mChacharaSeriesAdapter.loadNewData(chacharaSeriesList);
    }
}
```

En aquest codi es mostra com es crea una **Llista** en la que es van afegint tots els objectes sèrie(*ChacharaSeries*) i un cop estan tots es passen a l'"adaptador" per tal de que els mostri.

Per acabar, a aquest adaptador se li pot incorporar un "Listener" que sabrà quin d'aquests objectes ha clicat l'usuari i s'anirà a la pàgina desitjada.

```
mRecyclerViewSeries = (RecyclerView) findViewById(R.id.recycler_viewSeries);
mRecyclerViewSeries.setLayoutManager(new LinearLayoutManager(this));

mChacharaSeriesAdapter = new ChacharaSeriesAdapter(new ArrayList<ChacharaSeries>(),SelectSerieAD.this);
mRecyclerViewSeries.setAdapter(mChacharaSeriesAdapter);
mRecyclerViewSeries.addOnItemClickListener(new RecyclerView.OnItemClickListener() {
    @Override
    public void onItemClick(View view, int position) {
        Intent intent = new Intent(getApplicationContext(),SelectChapterActivity.class);
        intent.putExtra("CHACHARA_IDSERIE",mChacharaSeriesAdapter.getSerie(position).getId_serie());
        intent.putExtra("CHACHARA_AD", ad);
        startActivity(intent);
    }
});

@Override
public void onItemLongClick(View view, int position) {
}
});
```

Aquest procés es va repetint a totes les parts en què hi ha contingut dinàmic i no se sap quants objectes hi poden haver com per exemple amb els comentaris.

En el cas dels comentaris,també s'afegeix una altra funcionalitat, la del mostrar un diàleg quan s'avisava de que un comentari es SPAM. Per tal de mostrar el diàleg, l'usuari només cal que mantingui premut el TouchComentari en qüestió.

Per aconseguir-ho es fa servir el *DialogFragment*, el mateix que es fa servir a la pàgina de registre per seleccionar la data de naixement.

```

public class SpamButton extends DialogFragment {

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setMessage("Quieres marcar el mensaje como spam?")
            .setPositiveButton("Si", (dialog, id) -> {
                mListener.onDialogPositiveClick(SpamButton.this);
            })
            .setNegativeButton("No", (dialog, id) -> {
                mListener.onDialogNegativeClick(SpamButton.this);
            });

        return builder.create();
    }

    public interface NoticeDialogListener {
        public void onDialogPositiveClick(DialogFragment dialog);
        public void onDialogNegativeClick(DialogFragment dialog);
    }

    NoticeDialogListener mListener;

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        try {
            mListener = (NoticeDialogListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString()
                + " must implement NoticeDialogListener");
        }
    }
}

```

En el cas de seleccionar com a SPAM com que es una funció que s'utilitza en més d'una pàgina, les funcions depenent de què es seleccioni (SI/NO) es programen de forma externa a la pròpia pàgina.

```

//SPAM

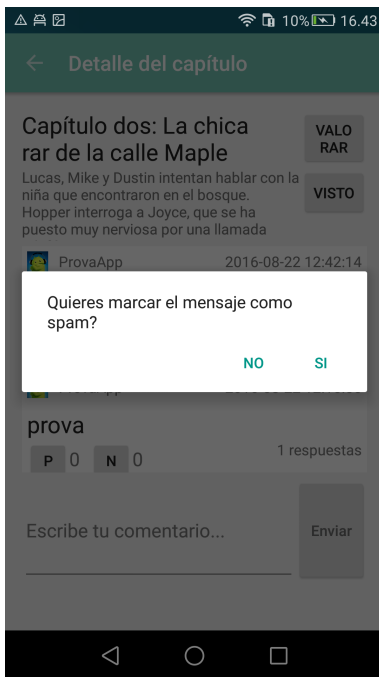
public void showNoticeDialog() {
    DialogFragment dialog = new SpamButton();
    dialog.show(getSupportFragmentManager(), "SpamButton");
}

@Override
public void onDialogPositiveClick(DialogFragment dialog) {
    System.out.println("El comentari: "+Integer.toString(comentariSPAM)+" es SPAM!");
    CommentSPAM commentSPAM = new CommentSPAM();
}

@Override
public void onDialogNegativeClick(DialogFragment dialog) {
    //RES A FER
}

```


Instantània de l'aplicació sobre el funcionament del **DialogFragment**:



Chachareo

En l'apartat de Chachareo s'utilitzen les implementacions que ja s'han anat explicant, El **DatePicker** (amb l'extensió també **HourPicker**), i el **RecyclerView** per a la selecció de les invitacions així com també quan surt la llista dels diferents *Chachareos*.

Per destacar en aquest apartat està el comportament del codi que s'encarrega del *Chachareo*. Primerament, l'aplicació crida el document **appChachareoMessages.php** que l'hi retorna tots els missatges publicats en aquell xat o *Chachareo*. Després registrem el timestamp (data/hora numèrica) i enviem aquest al cridar el php **appChachareoPush.php**. Aquest ara ens retornarà el següent missatge que s'afegeixi després de l'últim que ha rebut. Quan es rebi resposta es torna a repetir el procés d'agafar el timestamp i enviar-lo. Aquest procediment funciona perfectament en web però a l'hora d'implementar-ho es va veure que **no funcionava correctament amb android**, per a solucionar-ho i donat el poc temps del que es disposava es va decidir que l'aplicació crides cada 5 segons la primera pàgina. Si en un futur es volgués presentar la aplicació al públic s'hauria de buscar una alternativa.

Cercador

El funcionament del cercador en un principi estava determinat que poguessis cercar tant sèries com usuaris, però no s'ha sigut capaç de trobar la manera de fer-ho correctament així que tan sols cerca les sèries.

Primerament el cercador mostra una llista amb totes les sèries i quan l'usuari fa click a la icona li surt l'espai per tal de poder escriure el que vol cercar. Un cop ha acabat i envia la cerca se li mostren totes les sèries que coincideixen amb el nom o descripció del que ha cercat.

TFG: Creació d'una aplicació Android d'una xarxa social

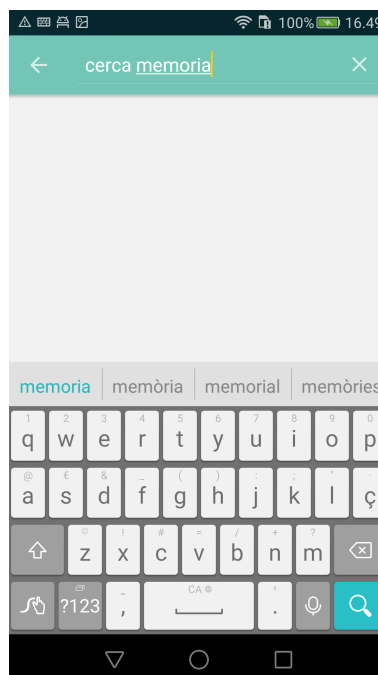
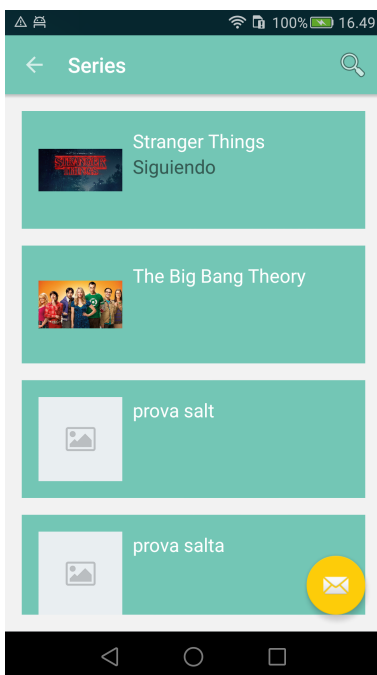
S'utilitza el **SharedPreferences** per passar la consulta, semblant al que es fa al registre i tot el tema de les animacions en la cerca les porta incorporades de sèrie l'Android utilitzant el **MenuInflater()**.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.pagina_principal, menu);
    final MenuItem searchItem = menu.findItem(R.id.menu_search);
    mSearchView = (SearchView) searchItem.getActionView();
    SearchManager searchManager = (SearchManager) getSystemService(Context.SEARCH_SERVICE);
    mSearchView.setSearchableInfo(searchManager.getSearchableInfo(getComponentName()));
    mSearchView.setIconified(false);
    mSearchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            SharedPreferences sharedPref = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
            sharedPref.edit().putString("CHACHARA_QUERY", query).commit();
            mSearchView.clearFocus();
            finish();
            return true;
        }

        @Override
        public boolean onQueryTextChange(String newText) {
            return true;
        }
    });
    mSearchView.setOnCloseListener(() -> {
        finish();
        return false;
    });
    return true;
}
```

CODI ENCARREGAT DE LA CERCA

Instantània de l'aplicació:

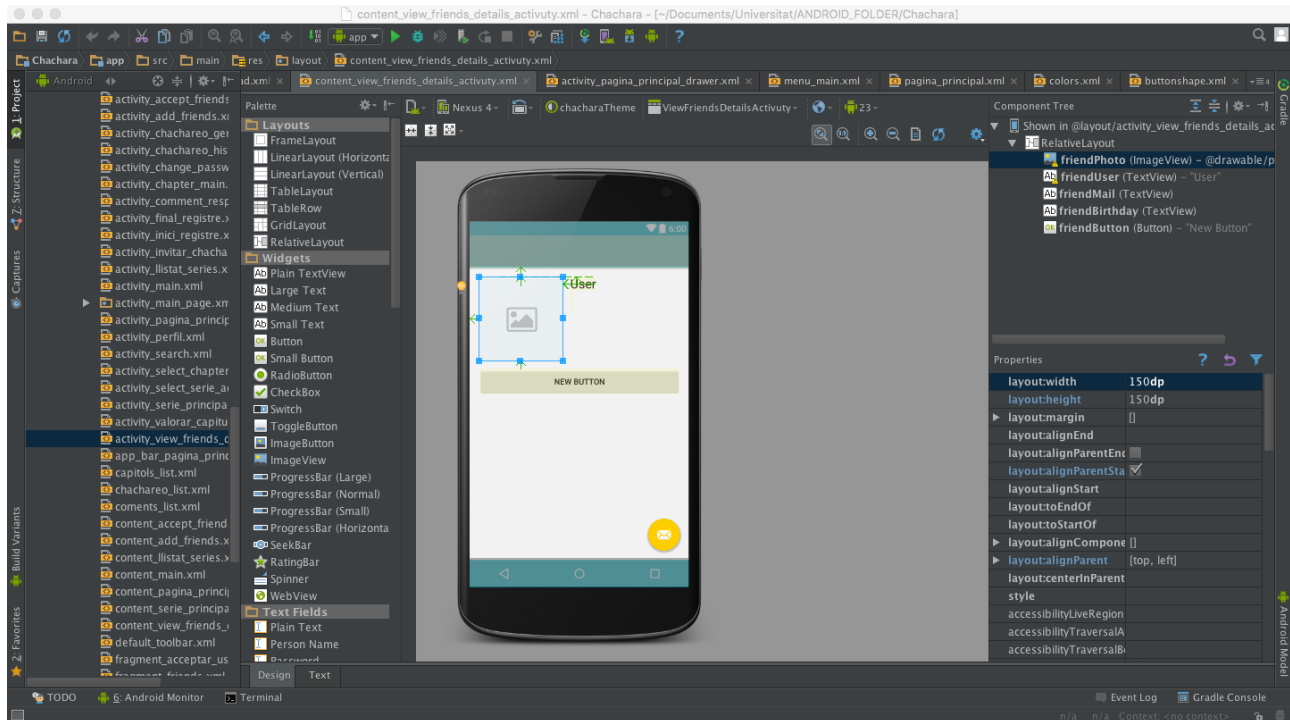


Programació estil visual i texts

Com funciona?

Layout

A Android l'apartat gràfic es programa en XML amb l'afegit que amb Android Studio tens un gestor gràfic però que no es massa fiable a l'hora de posicionar els elements. Tots els elements que tinguin un especte diferent als que porta android predeterminats s'han de crear amb un arxiu XML.



GESTOR GRÀFIC ANDROID STUDIO

La manera de funcionar de la part gràfica es jeràrquica i segueix les normes XML que per exemple limiten a que només hi hagi un contenidor "pare". Hi ha diversos tipus de contenidors : RelativeLayout, LinearLayout són els dos que es fan servir en aquesta aplicació i la diferència entre els dos es que el **LinearLayout** els objectes es van col·locant un sota l'altre (o al costat) segons poden i el **RelativeLayout** es fa per col·locar els objectes referenciant a altres objectes.

Altres

Com s'ha comentat, amb XML es configura tota la part gràfica. Es per això que per exemple el color té un XML (**colors.xml**) en el qual es troben els noms dels colors amb el seu codi hexadecimal. Al fitxer **styles.xml** és el document gràfic principal ja que s'hi configuren tots els colors de l'aplicació, els texts, la barra de navegació, etc...

Els menús també es troben programats en xml així com el botó flotant d'acció directe també és un xml a part. De la mateixa manera es programa i configura els textos i l'estil general de la app.

Els textos es troben localitzats en el fitxer **strings.xml** i son referenciats a aquesta localització durant tota l'estona i així es facilita la traducció de les apps.

Problemes dissenys

Ni que al finalitzar aquest treball hi ha els dissenys acabats poques son les pantalles en les quals es segueix el disseny realitzat per la meva companya Laura. No s'han pogut incorporar els dissenys ja que es van tenir problemes de coordinació amb els altres membres del treball i també perquè el desenvolupament d'aquest treball ha resultat ser d'una dificultat més alta de la que es va poder preveure en un primer moment. Es per això que el disseny de la aplicació es purament funcional i el proper pas abans de poder fer un hipotètic llançament al mercat seria l'adopció d'aquest disseny i arreglar qualsevol problema que en pogués esdevenir.

Testeig final

Al finalitzar la aplicació en la seva versió 1.0 es va procedir a realitzar una comprovació general de com seria el funcionament en un entorn real. Com es d'esperar ja que l'aplicació s'anava provant cada cop que es feien coses es veu que l'aplicació és funcional però que tal i com es detallava al principi es tracta d'un prototip i en cap cas estaria apte per entregar al públic general.

Primerament perquè el disseny no és el final i també perquè tal i com s'ha explicat amb anterioritat, el funcionament del xat és erràtic i no es va disposar de prou temps per arreglar-ho, ja que per a fer-ho s'havia de trobar un sistema alternatiu a l'efectuat.

Conclusions

En acabar aquest treball la sensació es agre-dolça, ja que per una part i la més important, almenys per a mi, es que s'ha pogut complir uns dels objectius principals i es que he pogut aprendre com es treballa amb Android, quines eines s'utilitzen i ara ja seria capaç de desenvolupar les aplicacions que em puguin venir al cap. També, s'ha aconseguit realitzar un prototip funcional bastant avançat del què seria l'aplicació final, però m'hauria agradat poder utilitzar el disseny final per tal de donar-li vida a l'aplicació més enllà del codi.

Com a objectius secundaris han estat la realització del treball en grup i posar a prova les metodologies de treball àgil en el que he pogut comprovar que són fàcils d'entendre però difícils de posar en pràctica i és una pena ja que si haguéssim estat una mica més acostumats a utilitzar aquest sistema probablement el resultat seria molt millor.

Per acabar, dir que em sento satisfet del resultat obtingut amb les excepcions comentades i donar gràcies al director del treball, en Juan José Fábregas perquè ens ha ajudat en tot el què hem necessitat i sobretot en l'aplicació dels mètodes de treball àgil.

Bibliografia

Artícles

LA OBSESIÓN ESPAÑOLA POR ANDROID I EL GADGETOBLOG I BLOGS I ELMUNDO.ES.
(N.D.). RETRIEVED MARCH 5, 2016, FROM [HTTP://WWW.ELMUNDO.ES/BLOGS/ELMUNDO/](http://www.elmundo.es/blogs/elmundo/)

EL-GADGETOBLOG/2015/05/08/LA-OBSESION-ESPANOLA-POR-ANDROID.HTML
REINA PÉREZ, M. (2014). ANÁLISIS DE LA AUDIENCIA DE LAS SERIES DE TELEVISIÓN A TRAVÉS DE LAS REDES SOCIALES: “BIENVENIDOS AL LOLITA” VS “EL PRÍNCIPE.”

Curs complert d'android

En la realització s'ha completat el curs d'Android del web per a cursos a distància udemy

AUTOR: TIM BUCHALKA

<https://www.udemy.com/master-android-7-nougat-java-app-development-step-by-step/>

Enllaços de suport

Android Show Dialog:

<https://developer.android.com/guide/topics/ui/dialogs.html>

Android floating menu:

<http://www.hermosaprogramacion.com/2016/02/floating-action-button-android/>

DatePickerDialog:

<http://android--examples.blogspot.com/2015/05/how-to-use-datepickerdialog-in-android.html>

Android TAB:

<http://www.androidhive.info/2015/09/android-material-design-working-with-tabs/>
<http://www.truiton.com/2015/06/android-tabs-example-fragments-viewpager/>

Android Button Maker

<http://angrytools.com/android/button/>

httpURLConnection:

<http://developer.android.com/reference/java/net/URLConnection.html>

Session id:

<http://stackoverflow.com/questions/11602457/session-id-passed-set-but-session-does-not-contains-the-variables-cookies-dis>

Push Chachareo (xat):

<https://www.youtube.com/watch?v=b6Z75EnIZTQ>